Cliplets: Juxtaposing Still and Dynamic Imagery

Neel Joshi Sisil Mehta Steven Drucker Eric Stollnitz
Hugues Hoppe Matt Uyttendaele Michael Cohen
Microsoft Research

ABSTRACT

We explore creating "cliplets", a form of visual media that juxtaposes still image and video segments, both spatially and temporally, to expressively abstract a moment. Much as in "cinemagraphs", the tension between static and dynamic elements in a cliplet reinforces both aspects, strongly focusing the viewer's attention. Creating this type of imagery is challenging without professional tools and training. We develop a set of idioms, essentially spatiotemporal mappings, that characterize cliplet elements, and use these idioms in an interactive system to quickly compose a cliplet from ordinary handheld video. One difficulty is to avoid artifacts in the cliplet composition without resorting to extensive manual input. We address this with automatic alignment, looping optimization and feathering, simultaneous matting and compositing, and Laplacian blending. A key user-interface challenge is to provide affordances to define the parameters of the mappings from input time to output time while maintaining a focus on the cliplet being created. We demonstrate the creation of a variety of cliplet types. We also report on informal feedback as well as a more structured survey of users.

INTRODUCTION

A taxonomy of visual imagery may begin with a separation of static images (photographs, paintings, etc.) from dynamic imagery (video, animation, etc.). A static photograph often derives its power by what is implied beyond its spatial and temporal boundaries, i.e., outside the frame and in the moments before and after it was taken. Our imagination fills in what is left out. Video loses some of that power, but being dynamic, has the ability to tell an unfolding temporal narrative. It carries us along through time.

In this work, we explore a category of media that in essence is more static image than video, but does contain some temporal elements. The media are derived from a short video, commonly only a few seconds. In particular, our focus is on results that juxtapose static and dynamic elements. Like a still photograph, the resulting piece of media can be "digested" in a short time interval on the order of 10 seconds.

An example of such media, referred to as *Cinemagraphs*, has recently appeared at a number of websites, displayed in the form of animated GIFs [6]. These cinemagraphs derive a visceral power by combining static scenes with a small repeating movement, e.g., a hair wisp blowing in the wind. Carefully staged and captured video coupled with professional tools like Adobe After Effects can result in well crafted cinemagraphs. Recently, at least three mobile based apps for creating cinemagraphs have also appeared.

Figure 1: Two cliplets. A looping layer on a still background (left). A combination of loop, mirror loop, and play layers (right). [Please see http://research.microsoft.com/cliplets/paper/ for a PDF with embedded videos that better conveys this result.]

At the cost of some confusion, we generalize from cinemagraphs and use a different term, *cliplet*, to denote media in which the dynamic elements do not need to be strictly looping. In addition, the juxtaposition of static and dynamic elements may be spatial (part of the frame is static while other parts are dynamic), temporal (a still followed in time by a dynamic element or vice versa), or both. The tension between these static and dynamic elements works to reinforce both elements within the cliplet.

Our paper makes a number of contributions in this area. First, we describe a set of idioms that characterize elements within cliplets. These idioms greatly simplify the process, relative to existing professional tools, yet provide significant expressiveness beyond the current set of mobile based apps.

These idioms are essentially mappings from the input video to layers of the result, and are used as building blocks in an interactive tool. The tool also provides simple user-driven segmentation, and automatic methods to aid in a variety of tasks, including image alignment, loop finding, spatiotemporal layer alignment and compositing, and photometric correction. As a result, our tool allows a wide range of cliplets to be easily produced from informally captured videos.

We also demonstrate a user interface that overcomes a key challenge of providing intuitive affordances for defining the parameters of the idioms that map input time to output time. We exercise the interface on a large number of input videos resulting in a collection of cliplets. In addition, we report

UIST'12, October 7-10, 2012, Cambridge, MA, USA.
Copyright 2012 ACM 978-1-xxxx-xxxx-x...$10.00.

on two informal evaluations of the cliplet application, based on extensive one-on-one demonstrations as well as a survey accompanying a publicly released version of the application.

RELATED WORK

Both the research literature and popular media include several examples of visual media that lie between a still and a video. A classic example is the animated GIF, originally created to encode short vector-graphics animations within a still image format. Another common example is simple panning and zooming over large static imagery, sometimes referred to as the Ken Burns Effect [16], which is often used in image slideshows.

More recently, the "Harry Potter" book and film series have popularized the notion that static images can also move. Perhaps the most relevant example, and one of our motivations in pursuing this work, is the "cinemagraph", a carefully crafted juxtaposition of still and moving image explored extensively by photographer Jamie Beck and designer Kevin Burg [6].

Until recently, the only widely available tools for creating cliplet-type imagery were general applications like Adobe Photoshop, Premiere, and After Effects. While these tools have the power to create cliplets, they are time-consuming to use in this context, and not easily accessible to an untrained user. Moreover they do not provide all the refinement operations necessary to overcome the inconsistencies and artifacts that commonly arise when constructing a cliplet. At least three mobile apps for cinemagraph creation have appeared (Cinemagr.am[1], Flixel[2], and Kinotopic[3]). Each of these allows a simple creation of a specific instance of a cinemagraph, a single looping feature within a still. Due to screen real-estate, and power constraints, none provide the generality, nor the artifact reduction techniques we describe. There are no technical descriptions available for these apps.

Many works in computer graphics and vision explore depicting either a static scene or short time sequence with some motion of the scene or observer: e.g., motion without movement [12], video textures [23, 17, 3, 21], animating stills [10], motion magnification [18], parallax photography [28], and spatiotemporal warping of video sequences [20].

Each of these examples adds a temporal element to what is essentially more like a static image than a video; however, none uses the same kind of static/dynamic tension to focus attention as is the emphasis in our work. We do, however, draw inspiration from these works, both in our technical approach and in developing a "language" and taxonomy for cliplets.

Specifically, video textures [23, 17] aim to create seamless video loops, where the entire spatial extent of a video is in motion. Research on panoramic video textures [3, 21] has a similar goal for videos spanning a large spatial extent. While the method of [3] may create a mixture of still imagery and video textures, this is a byproduct of an optimization rather than a desired intent, in the sense that the regions of still imagery correspond to mostly static regions in the input video. In our work the juxtaposition of still and moving imagery is an explicit goal, and our aim is to provide the user several forms of artistic control to extend and guide this process.

Work in video matting [9, 5] and video sprites [22] addresses the problem of extracting video regions, typically in a semantically meaningful way, for later compositing. In our setting, the primary segmentation goal is to find a boundary that forms a seamless composite in the resulting cliplet. In this respect, our work relates particularly to interactive digital photomontage [2], simultaneous matting and compositing [27], and video-to-still composition [14].

Perhaps the most closely related works are those of Tompkin et al. [26] and Bai et al. [4]. Tompkin et al.'s work takes a first step in the semi-automated creation of cinemagraphs. Their tool composites still and animated regions from a source video; it creates motion masks automatically and uses these to drive compositing. However, it offers few creative controls, mainly the ability to enable or disable looping per spatial region. The tool also has little provision to overcome or compensate for the many types of artifacts commonly encountered when creating cliplets. Bai et al.'s work focuses primarily on the challenging aspect of stabilizing sub-regions in the video. This is done with user-specified image-warping constraints. It does not address the concerns of developing an end-to-end system. Our contribution is to explore intuitive controls and several technical refinements that together enable users to quickly create high-quality results. Our work is also more general in that it develops a small set of idioms beyond simple looping and lets these idioms be sequenced together to develop short narratives.

THE LANGUAGE OF CLIPLETS

We define a cliplet as a 3D spatiotemporal object indexed by time t and 2D location x. A cliplet is formed by composing an ordered set of spatiotemporal output layers $\tilde{L}$, (we use tildes to denote the output), where each layer $\tilde{L}$ is mapped in time (and optionally space) from a corresponding input layer L within the given video. Input and output layers are subsets of the spatiotemporal volume of pixels in the input video and resulting cliplet. One key challenge is to provide the user with intuitive tools to map portions of the input video to the final cliplet.

We explore a number of mapping functions between the input and output layers. We begin by assuming that the *spatial* mapping from each input layer, $L(t, x)$, to output layer, $\tilde{L}(\tilde{t}, \tilde{x})$, is the identity. In other words, the effect of each layer is to modify some portion of the cliplet by overwriting each cliplet pixel using an input pixel from the same position, i.e., $x = \tilde{x}$, but offset in time.

Time-mapping idioms For each layer, the mapping from output to input time is defined by some function $t = \phi(\tilde{t})$:

$$\tilde{L}(\tilde{t}, \tilde{x}) = L(\phi(\tilde{t}), \tilde{x}).$$

This temporal mapping function, ϕ, characterizes the behavior of a layer and depends on various layer properties. In principle, an authoring system could provide direct, explicit control over ϕ, which is the approach used by professional tools such as Adobe Premiere, After Effects, and Photoshop.

[1] http://cinemagr.am/
[2] http://flixel.com
[3] http://kinotopic.com/

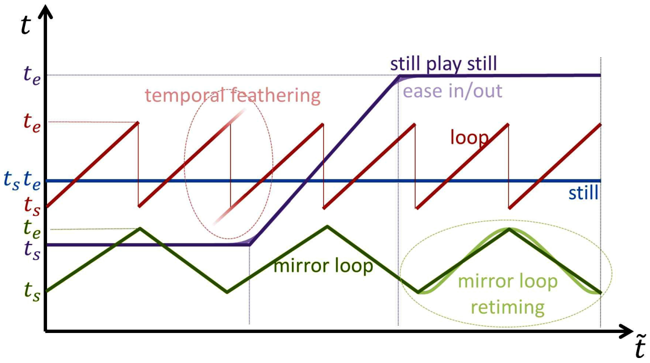


Figure 2: Illustration of various cliplet layer types, graphed using input video time t as a function of output cliplet time $\tilde{t}$. Subscripts s and e refer to the start and end times respectively. The diagram also illustrates layer refinements which are discussed in the Refinements section.

We initially developed our interactive tool to provide a similar type of direct control over the temporal mapping function, e.g. as a set of key-frames on a time in vs. out plot (as in Figure 2), allowing us to study the affordances of such an interface. After several months of use, we found that this explicit control was unnecessarily complex and made it difficult to perform common operations, like creating simple loops, for all but the most patient and highly experienced users.

Furthermore, we found that the typical mappings functions one needed to create cliplets, was actually quite limited. This led us to define a small set of iconic time-mapping idioms, (Still, Play, Loop, Mirror). Figure 2 shows $t = \phi(\tilde{t})$ for each idiom. As we will discuss in the Interactive Authoring System section, our current tool hides this complexity and instead uses simpler, higher-level controls to set the various layer properties.

The simplest idiom is a Still layer, in which the input layer, while spatially covering some or all of the full input, temporally consists of a only single video frame. The still time-mapping repeats this single frame to create an output layer that fills some portion of the cliplet volume. Typically, the first layer in a cliplet is a Still that spans the full spatial extent of the video, and is mapped over all frames of the cliplet. This initial layer forms a background layer over which additional layers are composed. These additional, more complex layers include spatial and temporal subregions of the input that are mapped in time via our set of idioms.

In our authoring system, layers are defined using the following attributes: an idiom (Still, Play, Loop, or Mirror), a spatial region $R(t)$, start and end times t_s, t_e within the input video, a playback velocity v relative to the input speed (default=1), start time $\tilde{t}_s$ within the cliplet, and optionally, an end time $\tilde{t}_e$ within the cliplet, otherwise the idiom is assumed to go on indefinitely.

A Still layer freezes a chosen input frame:

$$\phi_{\text{Still}}(\tilde{t}) = t_s, \quad \tilde{t}_s \leq \tilde{t} < \tilde{t}_e.$$

For a Play layer, we have

$$\phi_{\text{Play}}(\tilde{t}) = t_s + v\,(\tilde{t} - \tilde{t}_s), \quad \tilde{t}_s \leq \tilde{t} < \tilde{t}_s + (t_e - t_s)/v.$$

A Loop layer repeats a snippet of video multiple times:

$$\phi_{\text{Loop}}(\tilde{t}) = t_s + v\,((\tilde{t} - \tilde{t}_s) \bmod T),$$

where the loop period $T = (t_e - t_s)/v$.

Finally, a Mirror layer is a loop where the input is played successively forwards and backwards:

$$\phi_{\text{Mirror}}(\tilde{t}) = t_s + v\,\text{Hat}((\tilde{t} - \tilde{t}_s) \bmod 2T),$$

where $\text{Hat}(u)$ maps time to move forward for one interval T, and then backwards for another interval T:

$$\text{Hat}(u) = \begin{cases} u & \text{if } u \leq T, \\ 2T - u & \text{otherwise.} \end{cases}$$

INTERACTIVE AUTHORING SYSTEM

Four snapshots of our tool are shown in Figure 3. The tool is designed to be simple, yet expressive, in order to help users quickly create cliplets. The UI is the result of three primary design decisions: 1) the building blocks of a cliplet are controls corresponding to our set of idioms, 2) the main UI area shows both the input video and resulting cliplet in a single window, and is also used to define the layers' spatial boundaries, and 3) the UI uses two separate timelines, one for the input video at the top, and one for the output cliplet at the bottom.

The panel on the right has buttons to add new layers (Still, Loop, Mirror, and *compound* Still-Play-Still), including thumbnails of the layer spatial extents. A panel of advanced features (not shown) contains checkboxes to invoke refinements discussed in the Refinements section.

Associated with each timeline are representations of the temporal aspects of each layer. The three panels in the top of Figure 3 depict three states of an example cliplet creation session. Four layers are successively defined. The first layer is always a still background. It is indicated on the input timeline as a small blue mark that can be set by the user. This particular layer covers the full frame spatially, and also covers the full extent of the output timeline.

The second (yellow) layer (upper-left) shows the definition of a Still-Play-Still *compound* layer (SPS), which is really three individual layers laid consecutively in output time. The spatial extent has been indicated by drawing a selection region with a pencil tool. The input temporal extent is indicated by the yellow bar in the input timeline at the top. The user can drag the input start or end times independently or can drag its center to shift it in time. The layer is next positioned and sized on the output time slider. The wide yellow region indicates when this layer plays in the cliplet. The yellow bars to the left and right on the output indicate the duration of the stills before and after the play. If both are off, it is simply a single play layer. In the figure, the first frame is held as a still before the play layer, and the last frame is held as a still after the play layer.

The third (green) layer is a looping layer, indicated by the sawtooth line on the output timeline. The bright-green region defines the first temporal instance of the loop in the cliplet, while repeated instances are shown in mid-green color. The length of the loop and number of repetitions is set by the

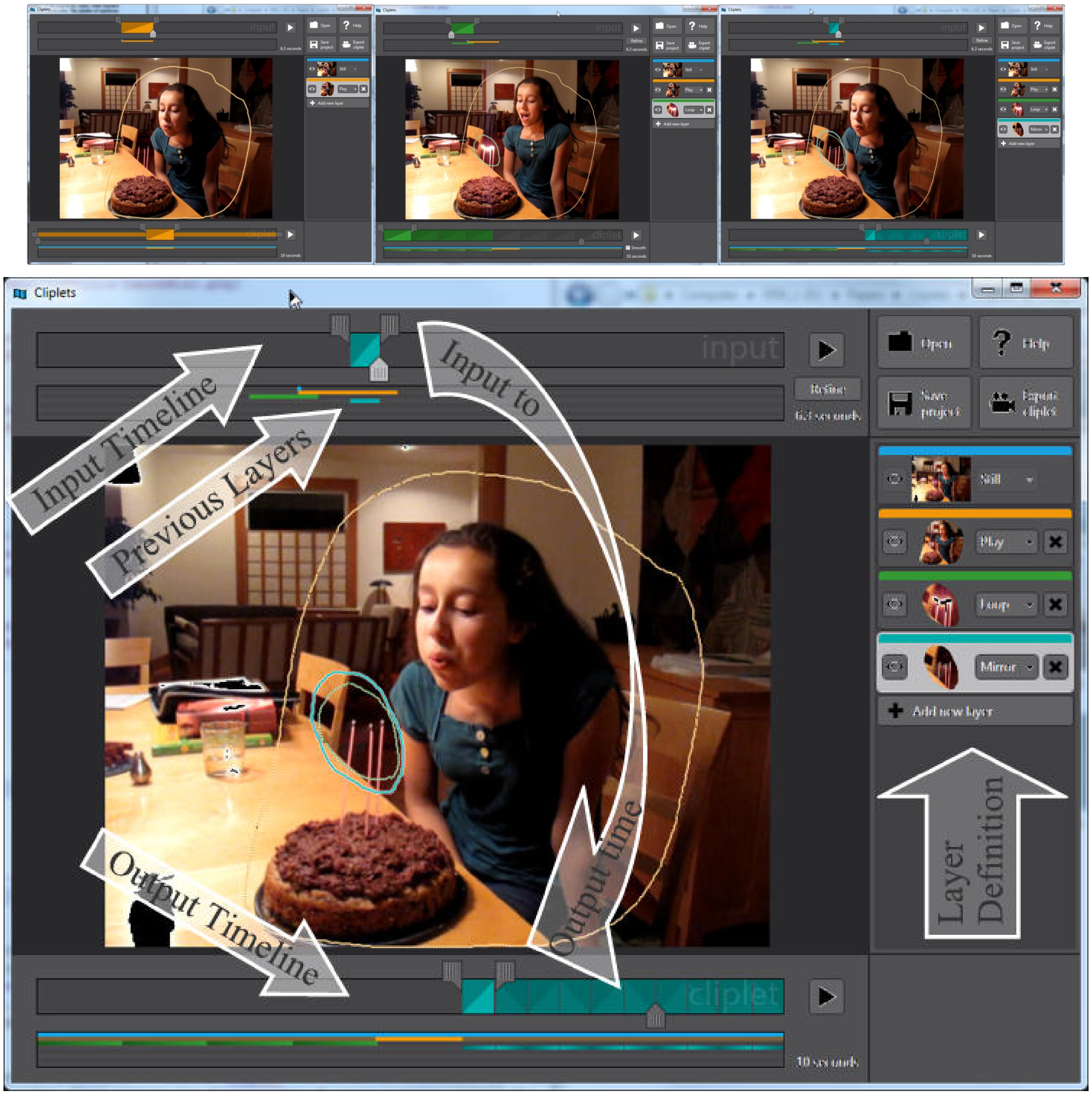


Figure 3: Interactive system for creating cliplets. The main area is used to indicate the spatial boundaries of layers and show the input video and resulting cliplet. The input and output timings for each layer are defined on the top and bottom timelines, respectively. The top three panels depict the process of defining a play layers, followed by a loop, and a mirror layer in sequence. [Please see http://research.microsoft.com/cliplets/paper/ for a video that shows the system in use.]

user. In this example, four instances are selected, thus this layer disappears from the cliplet at about the same time that the previously defined play layer starts. Note the indicators of previously defined layers depicted below the timelines. These act as guides to the user in defining subsequent layers. Since the looping layer's spatial extent overlaps that of the SPS, it is composited over it.

Finally, the fourth (blue) layer is a mirror loop. The up-and-down triangle line in the output timeline indicates each loop instance played forward and backward. In this example, the first mirror loop instance is positioned to begin at the midpoint of the output timeline, so it has no effect during the first half of the cliplet.

To see the system being used to create this cliplet, please visit http://research.microsoft.com/cliplets/paper/.

REFINEMENTS

A direct composition of layers can often fail to produce visually perfect cliplets. Handheld camera shake, as well as scene motion at layer boundaries, may reveal seams between layers. Changes in scene illumination or camera exposure can also create photometric seams at the boundaries. Temporal discontinuities created by the looping structure may reveal temporal seams.

We reduce these artifacts by refining the simple mappings (described in the Language of Cliplets section) between input video layers and the output cliplet. Specifically, we improve spatiotemporal continuity using a combination of techniques that (1) warp the layers both geometrically and temporally, (2) optimize the spatiotemporal region segmentation, and (3) blend the pixel colors at layer boundaries. These automatic refinements are presented to the user as a set of simple binary controls to either enable or disable the feature.

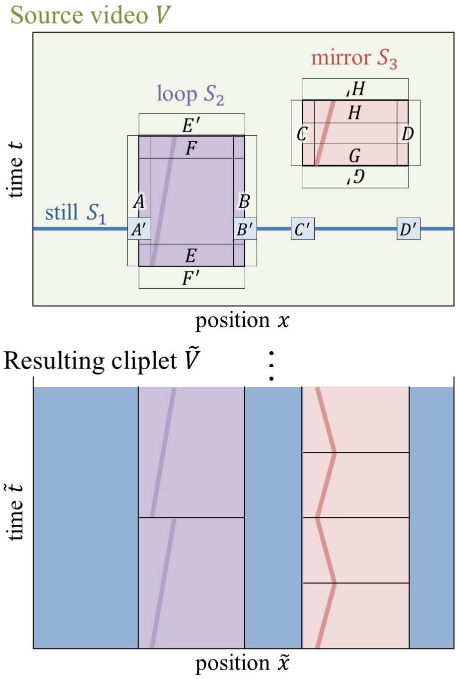


Figure 4: Three layers (Still, Loop, Mirror) access various spatiotemporal regions of the input video. For spatial continuity in the resulting cliplet, pixel colors in regions A, B, C, D should be similar to those in A', B', C', D' respectively. And for temporal continuity, pixel colors in regions E, F, G, H should be similar those in E', F', G', H' respectively.

Geometric refinement

We begin by modifying the spatial mapping from input to cliplet. Conceptually, we modify the spatial positions of the input pixels. Let us consider the illustration in Figure 4, which shows time-versus-position slices of the input video and output cliplet, in the presence of three layers (Still, Loop, and Mirror). The Still layer, S_1, sources its pixels from a single point in time, but spatially across the whole video. Small regions of the still, A' and B', abut a looping layer S_2. The goal of the geometric refinements is to make the corresponding regions in the looping layer, A and B, be as similar to A' and B' as possible, over the *entire time interval* $[t_s, t_e]$. Similarly, for the Mirror layer S_3, we wish C' and D' from the still to match C and D from the spatial boundary of the looping layer. Effectively, each pixel near a looping boundary should have constant color over the loop (i.e., zero optical flow). We vary the source location over time in the input video via global and local alignment operators to accommodate this goal.

Global alignment Because we allow the flexibility of working with handheld video, it is necessary to account for camera motion. As a preprocess, we use a modified video stabilization pipeline to align the video frames to simulate a still camera[13, 15]. The goal of the global alignment step is to find a sequence of similarity transforms, that when applied to the input removes all apparent camera motion (as if the camera had been on a tripod). To accomplish this, for every frame in the video we extract *Harris* features each with a corresponding *Brief* descriptor [8]. Between adjacent frames we do a windowed search to find matching features. The window size is dictated by the maximum frame-to-frame velocity that is expected in handheld videos. A feature is determined to be a match if the Brief descriptor distance of the best match is sufficiently different from that of the second best match (a.k.a. the ratio test[19]). This allows us to produce a set of feature tracks for the entire input video sequence.

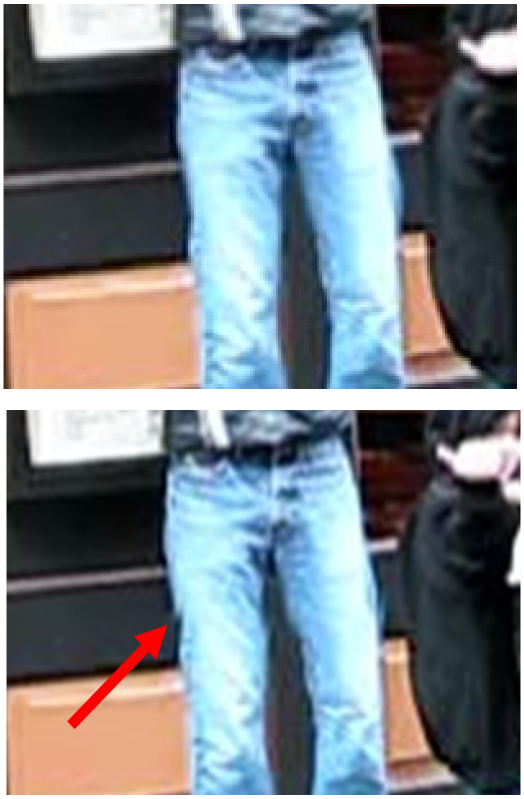


Figure 5: With local alignment (top) and without (bottom). Note that before the alignment the middle part of the person's leg tears noticeably on the left edge of his jeans.

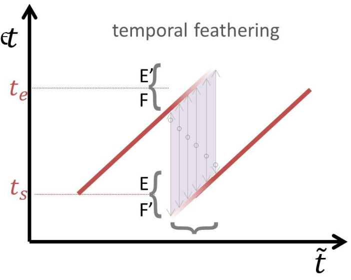
To avoid locking onto scene motion, the tracks are analyzed to distinguish foreground motion from background static features. The background feature tracks are assumed to belong to the largest set, such that a single temporal sequence of similarity transforms can map all background features back to their positions in frame 0. For this we employ a RANSAC (RANdom SAmple Consensus) [11] technique that runs over all of the frames simultaneously. The RANSAC iterations run by picking a random pair of tracks to determine a putative similarity transform $T[n, 0]$ between the last frame(n) and first frame(0) of the sequence. If $T[n, 0]$ maximizes the set of *inlier tracks*, then that same track pair is used to determine the transform $T[i, 0]$ between every other frame (i) and the first frame. For every frame (i) we test the current inlier tracks with $T[i, 0]$ and remove any inliers that are not within the RANSAC threshold. The set of feature tracks that maximizes the inlier count in this multi-frame sense are declared to be on the background and used to determine the transformations to stabilize the frames to simulate a still camera.

Local alignment Subtle scene motion, scene parallax, or small errors in global alignment can still cause spatial seams in a cliplet. For example, in Figure 5, one layer covers a man and his son but cuts though the man's legs. As the man rocks from side to side, a tear occurs at the layer boundary. We reduce this effect as follows. Within an eroded and dilated region near the layer boundary, R, we compute optical flow together with a confidence for the flow values based on local gradient magnitudes. Using a diffusion-based method for sparse data interpolation [24], these flow vectors are interpolated across the layer weighted according to their confidences values. We use the resulting smooth warp field to spatially deform the layer such that it aligns with the background at the boundaries.

Temporal refinement

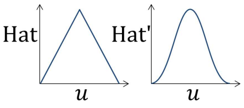
Much like the spatial discontinuities above, Loop and Mirror layers may introduce temporal discontinuities at their temporal boundaries, indicated as E, F, G, and H in Figure 4.

Optimized loop transitions As in Video Textures [23], for Loop layers we desire the frames just before and after the start and end frames to be as similar as possible. Thus, temporal artifacts are minimized if the temporal region E in Figure 4 matches the region E' just beyond the end of the layer and likewise F matches F'. Effectively, the endframes of the loop should be similar in both pixel colors and optical flow. To achieve this we seek start and end time pairs that minimize these differences. The user can let the system search the entire input video for the best candidate loops, or can specify an initial layer and have the system search nearby for improved start and end times.



Temporal feathering To further reduce Loop temporal discontinuities, we feather (i.e., morph) across the frame near the layer endframes. Over the small temporal regions (F', E) and (F, E'), we first use optical flow to warp E' to fit E, and call the result $\bar{E}$. Similarly, we create an $\bar{F}$ to fit F. Finally, we cross-fade between $\bar{F}$ and F and between E and $\bar{E}$ at the loop transition, as illustrated in the inset diagram, and as denoted by the oval marked "temporal feathering" in Figure 2. The effect can be seen in Figure 6.

Optimized mirror loop transitions For Mirror layers, the situation is quite different. We would like the small temporal regions around the start and end, G and H in Figure 4, to match their temporally inverted regions just beyond the temporal boundaries, G' and H'. This can only be true if the endframes have zero optical flow. We thus look for frames with little optical flow within the layer region. As with loop layers, we automatically present candidates near those specified, or the top set of candidates from the full input.



Slow-in/slow-out temporal mapping For Mirror layers any residual optical flow at the layer endframe results in jerky motion. We further reduce this type of artifact by slowing the motion as it approaches the layer endframes. For instance, in the case of Mirror layers, we replace $\mathrm{Hat}(u)$ by $\mathrm{Hat}'(u)$ as shown inset. Figure 2 indicates this in the oval marked "mirror loop retiming". We perform a similar retiming when transitioning between Still and Play layers as illustrated by the "ease-in ease-out" label in Figure 2. Because time is momentarily slowed considerably we generate new in-between frames. We again use optical flow to flow frames forward and backward and interpolate the results when the framerate drops below 10 fps.

Layer boundary refinement

Recall that to keep the UI simple, the user sketches a single region R for each layer at one selected time frame. Thus the layer's spatiotemporal extent $\tilde{L}_i$ is a generalized cylinder — the extrusion of the user-drawn region R over a fixed time interval $[t_s, t_e]$ obtained by the optimization in the Temporal Refinement section.

At times, the user may not completely encircle an object, or in subsequent frames, objects with significant motion may leave the boundary and/or unwanted objects may enter. In these cases, to improve spatiotemporal continuity, we can refine the extent $\tilde{L}_i$ in two ways. First, we compute optical flow between frames to advect the user-drawn boundary R to form a time-dependent region $R(t)$.

Second, we perform a binary graph-cut over the spatiotem-

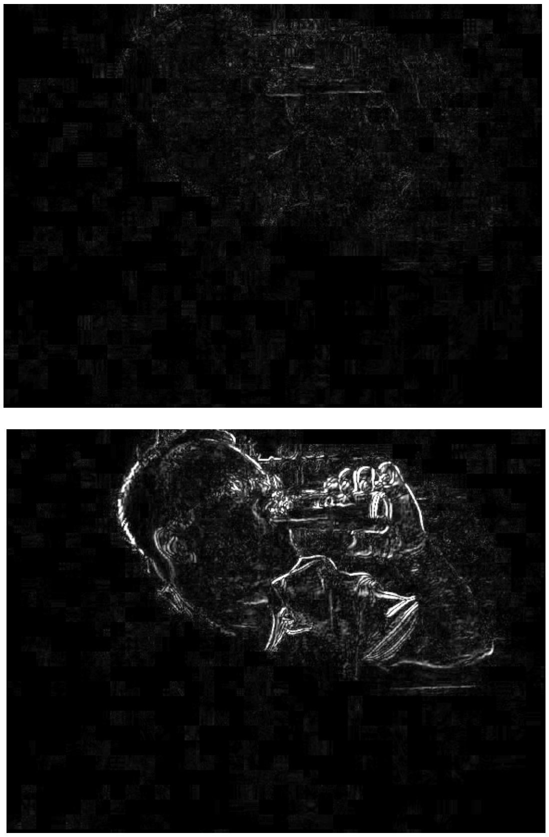


Figure 6: We show the difference between the start and end frame of a loop with temporal feathering (top) and without (bottom). Note that the temporal discontinuity is reduced in the feathered result, resulting in a more seamless loop. [Please see http://research.microsoft.com/cliplets/paper/ for a PDF with embedded videos that better conveys this result.]

poral volume to refine the layer boundary. The construction is similar that in graphcut textures [17]. It also leverages the observation in simultaneous matting and compositing [27] that, since both the layer and the background come from the same source material, a conservative matte that does not cut through the foreground tends to avoid artifacts.

The optimization is governed by a trimap computed from erosion and dilation on the region $R(t)$. Pixels between (and including) the two trimap boundaries are assigned nodes in the graph-cut formulation. A binary cut determines if the pixel is assigned to the new layer $\tilde{L}_i$ or retains its value from the reference image, depending on the spatiotemporal similarity with neighboring pixels. Thus the result of is a modified spatiotemporal extent $\tilde{S}_i$ whose boundaries adapt to both the source video content and the background over which it is composed. An example is in our video at http://research.microsoft.com/cliplets/paper/.

Blending during composition

To reduce any remaining spatial discontinuities, we perform per-frame Laplacian blending [7, 25] when compositing each layer. This reduces any artifacts due to changes in exposure times as well as small misalignments. In the inset comparison, the small gradient at the edge of the layer in the left panel causes a visual snap on each loop iteration which is removed in the right panel and in the resulting cliplet.

EVALUATION

The evaluation of the Cliplets tool has proceeded through two informal means. First, we demonstrated the tool one-on-one to approximately 1000 people during three days at a technical exhibition. This resulted in numerous online articles and blog entries. In addition, we released the tool to the public and recently requested users to voluntarily respond to a short survey about their experiences. We report informally on both personal observations and conversations at the exhibition and the more structured survey responses.

At the exhibition, we created Cliplets on-the-fly in a matter of seconds. The most obvious reaction was one of amazed joy as the juxtaposition of still and motion appeared. Often, we were then asked, “Can you make that part move (e.g., a bird), and freeze the rest?”. Most of the time we could satisfy such requests in a few seconds. The most common reason such intents could not be satisfied was due to overlapping motions. There were also some questions that indicated, that the 30 second demo did not make clear the differing roles of the input and output timelines. Such misconceptions could usually be clarified quickly with a second example.

The Cliplets application was placed online along with a few short tutorials of the tool in action, much like the demonstrations shown at the exhibition. The application asks users to optionally fill out a short survey after the second usage of the tool. As of this writing, over 60,000 people have downloaded the application. More recently, we added a voluntary survey, and over 400 people have responded. The online survey has helped us understand some of the experiences of those that have tried the Cliplets tool themselves (and were willing to respond to the survey).

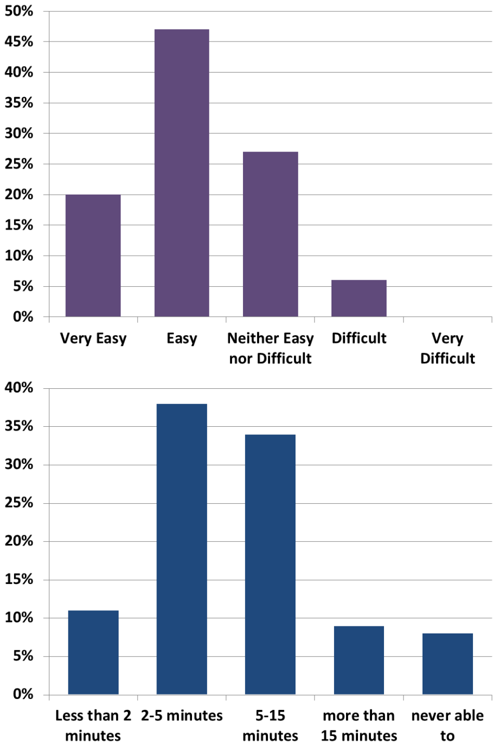


Figure 7: Ease of Use (top) and Completion Time (bottom)

The survey respondents were overwhelmingly male (90%), were fairly evenly spread in ages (8% under 20, 26% 20-39, 21% 30-39, 31% 40-59, and 14% 60 and over). Education levels were also well spread (4% completed elementary school, 23% high school, 46% undergraduate, and 27% graduate education). 75% had watched one or more tutorials. Only 11 found the tutorials “not useful”. The survey request was triggered after the second time users were about to close the application. At that point, about half had made 2 or more cliplets. About 1/3rd had made only one cliplet and 17% reported making none.

We asked how easy or hard it was to create a cliplet on a 5-point Likert scale. About two thirds found it easy (47%) or very easy (20%), with most of the rest (27%) reporting it was neither easy nor difficult. Only 6% reported it difficult and one respondent reported it being very difficult (see Figure 7(top)). When asked how long it took to create a cliplet they were satisfied with, the most common answer was 2-5 minutes (37%) with 11% reporting success in less than 2 minutes. 34% were able to create a satisfactory cliplet in 5-15 minutes, 9% took more than 15 minutes, and 8% were unable to create a satisfactory cliplet (see Figure 7(bottom)).

We also asked about understanding and usage of some of the features. All but 5% reported a basic or good understanding of the timelines. All but 5% reported understanding the layers concept. Users indicated using 2 layers in a cliplet as the most common, with a quarter creating cliplets with more than 2 layers. Loops were the most used layer type, however each of the other idioms was reported having been used by at least 30% of respondents. Input videos were derived from video cameras (66% indicating this source), mobile devices (27%), as well downloaded videos (26%), with a few indicating other sources. Interestingly, about half of respondents said they had shot videos specifically with making a Cliplet in mind. Half also indicated having shared a resulting cliplet, split almost evenly between using email and posting on a sharing site.

We also asked for freeform comments to indicate particular aspects of the system they liked and disliked, as well as features they would like to see added. There were a large number of very complimentary comments, perhaps our favorite being, "It's very simple to use, I like that very much. So simple I'm not too worried about my dad downloading this to play about and asking me too many questions." There were a number of requests for more advanced features such as being able to specify the changing shape of the mask through time, and to be able to combine layers from more than one video. Professional tools do provide this but at a cost to the simplicity of the interface. This type of tension is an expected one and perhaps impossible to overcome. The most common request was to create GIF output instead of mp4's. We recently added this ability in a new version.

RESULTS AND DISCUSSION

We have created numerous cliplets using our tool. Most of the source material comes from casual handheld videos shot by the authors. Depending on the complexity of the cliplet, they took from 2 to 10 minutes to construct. A number of examples are shown in Figure 8. We demonstrate cliplets created with combinations of four idioms: Still, Play, Loop, and Mirror. Our results show the interplay of these idioms resulting in a range of imagery from cinemagraphs to short narratives. [Our complete set of video results, including those from external users, can be found at http://research.microsoft.com/cliplets/paper/.]

Automation vs. Interaction The system we describe gives the user creative control over the semantic elements of the cliplets. The refinements automate minor adjustments to clean up the spatiotemporal boundaries of layers. In a few instances, however, the automated refinements change the intended semantics of the cliplet. For example, spatial adjustments may enlarge the layer to include unwanted foreground elements. Adjustments to temporal boundaries may result in a smoother loop transition, but may remove something the user intended to have in the layer. This tension between automatic and interactive operations is not uncommon in many semi-automated systems. In these cases, the user is able to turn off the refinements but may need to manually make precise adjustments to the layer boundary.

Failure cases There were a few common types of failure cases that do not arise in professionally created cinemagraphs. One failure mode occurs when the video could not be aligned due to excessive camera or foreground motion nor when there is no background to lock on to. Another difficult case arises when the object of interest overlaps a moving background. In some cases this could be handled by creating a "clean plate", which is something we are pursuing as future work; however, there are times where there is no possibility of constructing a clean plate since some of the background is revealed only during the loop. The boundary refinement can also fail if the user specified boundary is too far away from the "ideal" boundary, as the trimap will not contain a good region for creating a lower cost boundary. Boundary refinement in its most general form is equivalent to video segmentation and matting, an area of continued active research, thus there are numerous opportunities for future work here. Given these constraints, we were pleasantly surprised how often a successful cliplet could be derived from spontaneously and casually captured video.

SUMMARY AND FUTURE WORK

We have presented a tool for creating cliplets – a type of imagery that sits between stills and video, including imagery such as video textures and "cinemagraphs". We have identified a few simple time-mapping idioms that, when combined, provide a wealth of freedom for creativity. Our tool allows cliplets to be quickly created from casual handheld video using simple interactions. Numerous examples are provided.

Further analysis We made three fundamental design decisions in our work: 1) creating a reduced set of mapping functions that does not allow full control yet is more expressive than the controls in related mobile apps, 2) creating separate in and out timelines instead of a single time-in vs. time-out graph, and 3) using one pane instead of two for displaying the input video and output cliplet. While these choices seem valid given our experiences and initial user feedback, they do depart somewhat from previous work and each choice warrants more formal study, which is an interesting direction for future work.

A new way to see When one picks up a camera and heads into the world to photograph it, one often *sees* the world differently. One sees shots to frame, and instants to grab, rather than buildings, people, and traffic. Ansel Adams describes *seeing the finished photo in your mind's eye* [1]. In a similar way, after using our tool, we began to *see* the world through this new type of lens when thinking about capturing video. Small motions became things to focus on while the rest of the busy world faded away. This, as much as the results, has convinced us that cliplets represent an exciting and provocative new media type, and that intuitive tools for creating such media can be very powerful.

ACKNOWLEDGEMENTS

We thank Bill Freeman for initial discussions.

REFERENCES

1. A. Adams. *Camera*. Number v. 1. Houghton Mifflin Harcourt P, 1980.
2. Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3):294–302, 2004.
3. Aseem Agarwala, Ke Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Richard Szeliski. Panoramic video textures. *ACM Trans. Graph.*, 24(3):821–827, July 2005.
4. Jiamin Bai, Aseem Agarwala, Maneesh Agrawala, and

Ravi Ramamoorthi. Selectively de-animating video. *ACM Transactions on Graphics*, 31(4), 2012.

5. Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Trans. Graph.*, 28(3), 2009.

6. Jamie Beck and Kevin Burg. Cinemagraphs. http://http://cinemagraphs.com/.

7. Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 2(4):217–236, October 1983.

8. Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer vision: Part IV*, ECCV'10, pages 778–792, 2010.

9. Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski. Video matting of complex scenes. *ACM Trans. Graph.*, 21(3), 2002.

10. Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H. Salesin, and Richard Szeliski. Animating pictures with stochastic motion textures. *ACM Trans. Graph.*, 24(3):853–860, 2005.

11. Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

12. William T. Freeman, Edward H. Adelson, and David J. Heeger. Motion without movement. In *SIGGRAPH Proceedings*, pages 27–30, 1991.

13. Michael L. Gleicher and Feng Liu. Re-cinematography: Improving the camerawork of casual video. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5(1):2:1–2:28, October 2008.

14. Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of ACM Symposium on User Interface Software and Technology*, UIST '08, pages 3–12, 2008.

15. Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust L1 optimal camera paths. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, 2011.

16. Randy Kennedy. The still-life mentor to a filmmaking generation. The New York Times, October 6, 2006.

17. Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, July 2003.

18. Ce Liu, Antonio Torralba, William T. Freeman, Frédo Durand, and Edward H. Adelson. Motion magnification. *ACM Trans. Graph.*, 24(3):519–526, July 2005.

19. David G. Lowe. Object recognition from local scale-invariant features. *IEEE International Conference on Computer Vision*, 2:1150, 1999.

20. Alex Rav-Acha, Yael Pritch, Dani Lischinski, and Shmuel Peleg. Spatio-temporal video warping. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, 2005.

21. Alex Rav-Acha, Yael Pritch, Dani Lischinski, and Shmuel Peleg. Dynamosaicing: Mosaicing of dynamic scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(10):1789–1801, October 2007.

22. Arno Schödl and Irfan A. Essa. Controlled animation of video sprites. In *ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 121–127, 2002.

23. Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In *SIGGRAPH Proceedings*, pages 489–498, 2000.

24. Richard Szeliski. Locally adapted hierarchical basis preconditioning. *ACM Trans. Graph.*, 25(3):1135–1143, 2006.

25. Richard Szeliski, Matt Uyttendaele, and Drew Steedly. Fast Poisson blending using multi-splines. In *IEEE International Conference on Computational Photography*, April 2011.

26. James Tompkin, Fabrizio Pece, Kartic Subr, and Jan Kautz. Towards moment images: Automatic cinemagraphs. In *Proceedings of the 8th European Conference on Visual Media Production (CVMP 2011)*, November 2011.

27. Jue Wang and Michael F. Cohen. Simultaneous matting and compositing. In *IEEE Computer Vision and Pattern Recognition*, pages 1 –8, June 2007.

28. Ke Colin Zheng, Alex Colburn, Aseem Agarwala, Maneesh Agrawala, David Salesin, Brian Curless, and Michael F. Cohen. Parallax photography: creating 3D cinematic effects from stills. In *Proceedings of Graphics Interface*, pages 111–118, 2009.

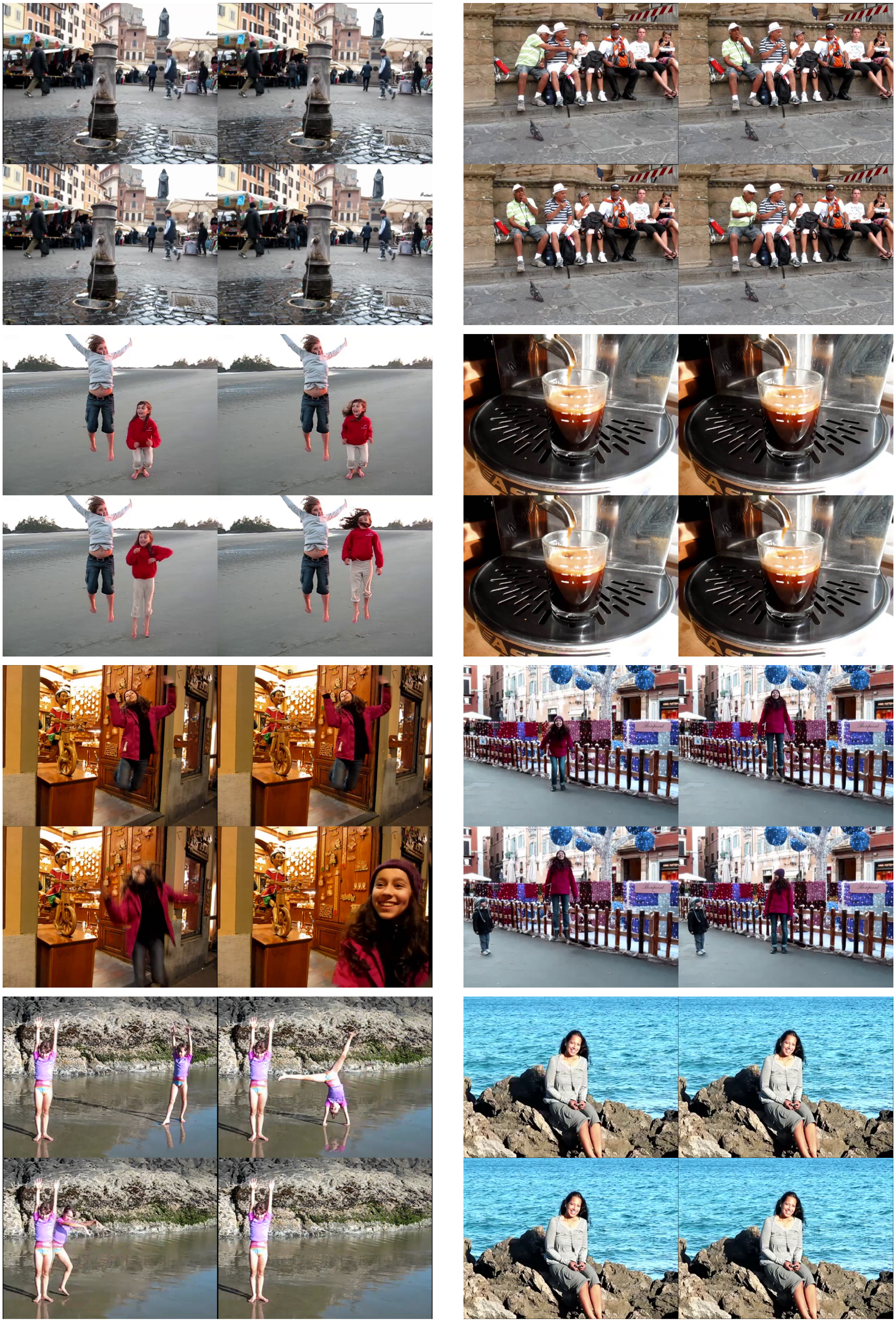


Figure 8: A variety of cliplets. Our tool has been used to create numerous cliplets. Each cliplet is a juxtaposition of at least two of our four idioms: Still, Play, Loop, and Mirror, and represent imagery from “cinemagraphs” to short narratives. [Please see `http://research.microsoft.com/cliplets/paper/` for a PDF with embedded videos that better conveys this result.]